

## **CLAIMS**

What is claimed is:

1. A method, comprising:

monitoring a state of an application running in a system, wherein said monitoring the state of the application includes monitoring one or more buffers associated with the application;  
controlling dispatch of one or more threads in the system, and wherein at least one thread in the system is associated with the application; and  
managing resources in the system based at least on the state of the application and the state of the one or more threads in the system.

2. The method of claim 1, wherein a thread includes one or more activities, and wherein said controlling the dispatch of the one or more threads in the system includes assessing execution readiness of the one or more activities.

3. The method of claim 2, wherein said controlling the dispatch of the one or more threads in the system includes delaying a ready-to-be-dispatched activity from being dispatched.

4. The method of claim 3, wherein a first activity is delayed from being dispatched to wait for a second activity to be ready so that both the first and second activities can be dispatched together, and wherein the first and second activities are from one or more applications.

5. The method of claim 1, further comprising:  
monitoring a machine state of the system, wherein said monitoring the machine state includes:  
determining resources available in the system, and  
increasing or decreasing the resources available in the system based on the state of the application and the state of the one or more threads in the system.
6. The method of claim 5, wherein the resources include configurable hardware components.
7. The method of claim 6, wherein the configurable hardware components include one or more processors, hardware buffers, memory, cache, arithmetic logic unit (ALU), and registers in the system.
8. The method of claim 7, wherein said increasing or decreasing the resources available in the system includes configuring the frequencies applied to at least the one or more processors in the system.
9. The method of claim 8, wherein said increasing or decreasing the resources available in the system includes configuring the voltages applied to at least the one or more processors in the system.
10. The method of claim 7, wherein said increasing or decreasing the resources in the system includes powering on or powering off at least a portion of circuitry in the system.

11. The method of claim 1, wherein said monitoring the one or more buffers associated with the application includes monitoring buffer fullness levels of the one or more buffers.

12. The method of claim 11, wherein said monitoring the buffer fullness levels includes, for each buffer associated with the application, comparing a buffer level with predetermined buffer fullness levels, wherein the predetermined buffer fullness levels include a high level mark and a low level mark.

13. The method of claim 12, wherein said comparing is to determine buffer overflow and buffer underflow conditions.

14. A computer readable medium containing executable instructions which, when executed in a processing system, causes the processing system to perform a method comprising:

monitoring a state of an application running in a system, wherein said monitoring the state of the application includes monitoring buffer fullness levels of one or more buffers associated with the application;

controlling dispatch of one or more threads in the system, and wherein at least one thread in the system is associated with the application; and

managing resources in the system based at least on the state of the application and the state of the one or more threads in the system.

15. The computer readable medium of claim 14, wherein said controlling the dispatch of the one or more threads in the system includes delaying a ready-to-be-dispatched thread from being dispatched.

16. The computer readable medium of claim 14, further comprising:  
monitoring a machine state of the system, wherein said monitoring the machine state includes:

determining resources available in the system, and

increasing or decreasing the resources available in the system based on the state of the application and the state of the one or more threads in the system.

17. The computer readable medium of claim 16, wherein the resources include configurable hardware components.

18. The computer readable medium of claim 16, wherein said increasing or decreasing the resources available in the system includes configuring the frequencies and/or the voltages applied to at least the one or more processors in the system.

19. The computer readable medium of claim 14, wherein said monitoring the buffer fullness levels includes, for each buffer associated with the application, comparing a buffer level with predetermined buffer fullness levels, wherein the predetermined buffer fullness levels include a high level mark and a low level mark.

20. A method, comprising:

managing resources in a system by monitoring buffer fullness levels of one or more buffers used by an application running in the system and by coordinating dispatch of threads in the system.

21. The method of claim 20, wherein said monitoring the buffer fullness levels of the one or more buffers includes monitoring buffer level of each of the one or more buffers to determine buffer overflow or underflow conditions.

22. The method of claim 20, wherein said coordinating the dispatch of the threads in the system is performed to increase overlap in threads execution.

23. The method of claim 22, wherein increasing the overlap in the threads execution includes changing a thread from a ready state to a queued state.

24. The method of claim 23, wherein the thread remains in the queued state until there is another thread in the ready state such that both threads can be dispatched together.

25. The method of claim 20, wherein said coordinating the dispatch of the threads in the system includes determining threads dependency, and wherein when there is dependency between a current thread and a next thread of the application and the buffer fullness levels of all of the buffers indicate a potential overflow condition, the resources in the system are increased.

26. The method of claim 20, wherein when the buffer fullness levels of all of the buffers reach a critical stage, the resources in the system are adjusted.

27. A computer readable medium containing executable instructions which, when executed in a processing system, causes the processing system to perform a method comprising:

managing resources in a system by monitoring buffer fullness levels of one or more buffers used by an application running in the system and by coordinating dispatch of threads in the system.

28. The computer readable medium of claim 27, wherein said monitoring the buffer fullness levels includes monitoring for buffer overflow or underflow conditions, and correspondingly increasing or decreasing the resources in the system to avoid the buffer overflow or underflow conditions.

29. The computer readable medium of claim 27, wherein said coordinating the dispatch of the threads in the system includes changing a thread from a ready state to a queued state and dispatching threads to increase execution overlap.

30. A method, comprising:

monitoring states of one or more threads in a system,

wherein said monitoring the states of the one or more threads includes

monitoring readiness to dispatch of the one or more threads; and

managing resources in the system based at least on the states of the one or more threads in the system, wherein the resources include configurable hardware

components in the system.

31. The method of claim 30, wherein said managing the resources in the system includes configuring the frequencies and/or voltages applied to at least one or more processors in the system.

32. The method of claim 30, wherein said managing the resources in the system further includes powering on or powering off a portion of circuitry in the system.

33. The method of claim 30, further comprising:  
managing state of an application running in the system, wherein the state of the application includes buffer fullness levels of one or more buffers used by the application, and wherein the resources in the system is increased or decreased based on the state of the application.

34. A method, comprising:  
monitoring a state of an application running in a system,  
    wherein said monitoring the state of the application includes monitoring buffer  
    fullness levels of one or more buffers associated with the application; and  
managing resources in the system based on the state of the application.

35. The method of claim 34, further comprising managing execution readiness of one or more threads in the system, and wherein said managing the resources is further based on the execution readiness of at least a thread associated with the application.

36. The method of claim 35, wherein the execution readiness includes a ready-to-be-dispatched state and a delay-from-being-dispatched state.

37. The method of claim 36, wherein said managing the execution readiness of the one or more threads in the system includes placing a thread from a ready-to-be-dispatched state into a delay-from-being-dispatched state.

38. The method of claim 37, wherein said monitoring the buffer fullness levels of the one or more buffers includes:  
determining a potential buffer underflow or overflow condition; and  
configuring one or more components in the system to change resources available in the system to avoid the potential buffer underflow or overflow condition.

39. A system, comprising:  
a memory to store data and instructions;  
a processor coupled to said memory on a bus, said processor operable to perform instructions, said processor comprising:  
    a bus unit to receive a sequence of instructions from said memory;  
    an execution unit coupled to said bus unit, said execution unit to execute said sequence of instructions, said sequence of instructions to cause said execution unit to:  
        monitor a state of an application running in a system, wherein said monitoring the state of the application includes monitoring buffer fullness levels of one or more buffers associated with the application;



control dispatch of one or more threads in the system, and wherein at least one thread in the system is associated with the application; and manage resources in the system based at least on the state of the application and the state of the one or more threads in the system.

40. The system of claim 39, wherein said controlling the dispatch of the one or more threads in the system includes delaying a ready-to-be-dispatched thread from being dispatched.

41. The system of claim 39, further comprising instructions to cause said execution unit to monitor a machine state of the system, wherein said monitoring the machine state includes:  
determining resources available in the system, and  
increasing or decreasing the resources available in the system based on the state of the application and the state of the one or more threads in the system.

42. The system of claim 41, wherein the resources include configurable hardware components.

43. The system of claim 41, wherein said increasing or decreasing the resources available in the system includes configuring the frequencies and the voltages applied to at least the one or more processors in the system.

44. A system, comprising:  
a multi-threading processor; and

a resource manager coupled to the multi-threading processor, the resource manager is to monitor states of an application running in the system, the states of the application including buffer fullness levels of one or more buffers used by the application, the resource manager is to further monitor states of one or more threads in the system for execution readiness, wherein the resource manager is to increase or decrease resources available in the system depending on the state of the application and/or the states of the one or more threads in the system.

45. The system of claim 44, wherein the resource manager is to change the execution readiness of a thread from a ready state to a queued state to increase subsequent thread execution overlap with execution of another thread.

46. The system of claim 45, wherein the resource manager is to change the execution readiness of a thread from a ready state to a queued state to increase subsequent system idle time when there is no thread execution.

47. The system of claim 46, wherein the resource manager is to increase or decrease the resources available in the system to avoid buffer underflow or overflow conditions to occur to the one or more buffers.

48. An apparatus, comprising:

logic to monitor states of an application running in a system, the states of the application including buffer fullness levels of one or more buffers used by the application;

logic to monitor states of one or more threads in the system for execution readiness; and

logic to adjust resources available in the system depending on the state of the application and/or the states of the one or more threads in the system.

49. The apparatus of claim 48, further comprising:

logic to change the execution readiness of a thread from a ready state to a queued state when it is determined that there is no other thread running or ready to be dispatched.

50. The apparatus of claim 48, wherein the logic to adjust the available resources in the system includes logic to determine if the buffer fullness levels of one or more buffers are in a critical stage.